Banquet Talk: Area-Time-Power tradeoffs in computer design: the road ahead

Michael J. Flynn

Single Stage Fehskens-Malewicki Equations:

burnout velocity:

$$v_b = \sqrt{\frac{F-mg}{k}} \tanh\left[\frac{t_b}{m}\sqrt{k(F-mg)}\right]$$

burnout altitude:

$$y_b = \frac{m}{k} \ln\left\{\cosh\left[\frac{t_b}{m}\sqrt{k(F-mg)}\right]\right\}$$

coast altitude:

$$y_c = \frac{m_b}{2k} \ln\left[\frac{k v_b^2}{m_b g} + 1\right]$$

coast time:

$$t_c = \sqrt{\frac{m_b}{g\,k}} \tan^{-1}\left[v_b \sqrt{\frac{k}{g\,m_b}}\right]$$

Where:

$k = \frac{1}{2}\rho C_D A$

ρ = atmospheric density

C_D = drag coefficient

A = frontal area

t_b = burn time

F = average thrust

m = average thrusting mass

m_b = burnout mass

g = acceleration due to gravity

Return

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 FEB 2005	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Banquet Talk: Area-Time-Power tradeoffs in computer design: the road ahead	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stanford University**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**33**	

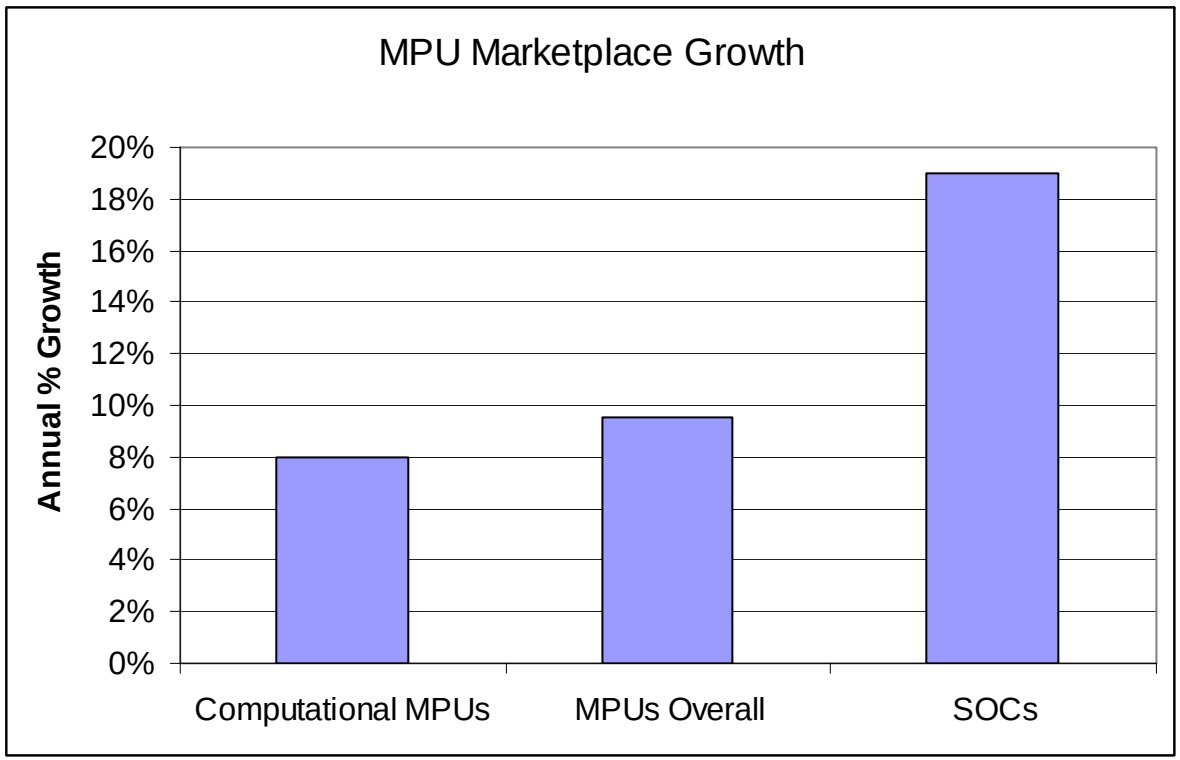
The marketplace

The computer design market (in millions produced, not value)

Device Type	2001	2002	2003	2004	2005	2006
Computational MPUs	147	151	168	187	202	218
Embedded MPUs	159	152	161	167	177	186
4- and 8-bit MCUs	3982	4180	4570	5090	5540	5970
16-bit MCUs	589	626	678	714	797	797
32-bit MCUs	190	216	247	368	511	673
DSP	484	526	569	638	694	691
Total	5551	5851	6393	7164	7921	8535

REF:J. Hines, "2003 Semiconductor Manufacturing Market: Wafer Foundry." Gartner Focus Report, August 2003.

The computer design market growth



B. Lewis, "Microprocessor Cores Shape Future SLI/SOC Market." Gartner, July 2002

Moore's Law

- Something doubles (or is announced to double) right before a stockholders meeting.
- So that it really doubles about every 2 years

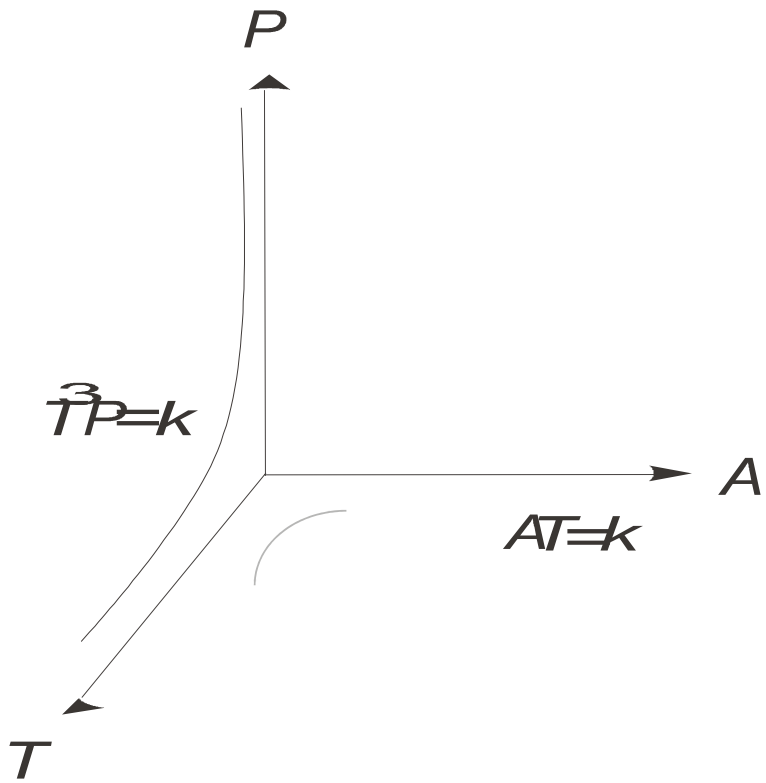
Suggested by: Greg Astfalk lecture

Semiconductor Industry Roadmap

Semiconductor Technology Roadmap

Year	**2004**	**2007**	**2013**	**2016**
(f) Technology generation (nm)	90	65	32	22
Wafer size (cm)	30	30	45	45
(ρ) Defect density (per cm^2)	0.14	0.14	0.14	0.14
(A) μP die size (cm^2)	3.1	3.1	3.1	3.1
!!! Chip Frequency (GHz)	4.2	9.3	23	39.6
MTx per Chip (Microprocessor)	553	1204	4424	8848
!!! MaxPwr(W) High Performance	158	189	251	288

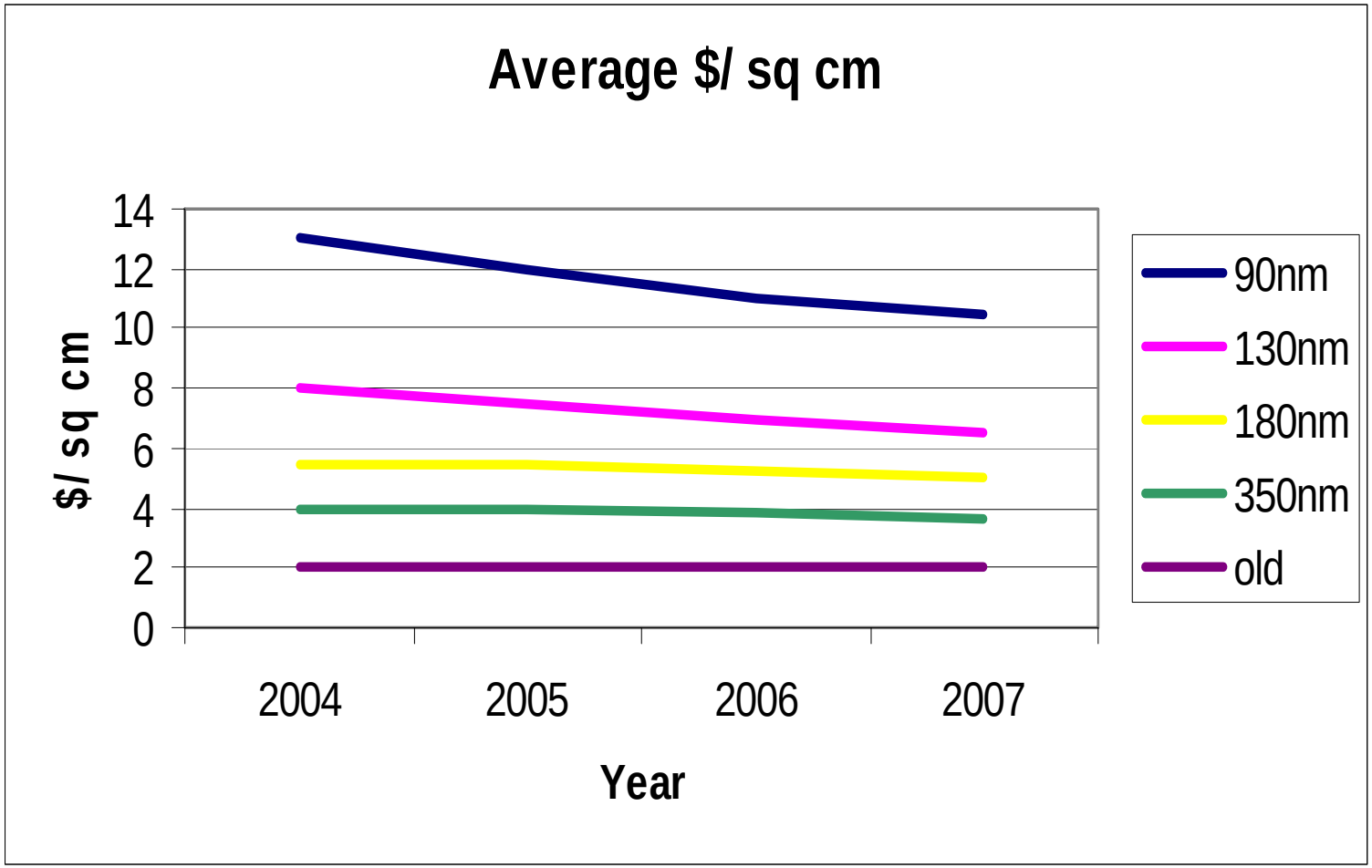
Time (Performance), Area and Power Tradeoffs



There's a lot of silicon out there … and it's cheap

www.newburycamraclub.org.uk

Cost of Silicon



REF:J. Hines, "2003 Semiconductor Manufacturing Market: Wafer Foundry." Gartner Focus Report, August 2003.

The basics of wafer fab

- A 30 cm, state of the art ($\rho = 0.2$) wafer fab facility might cost $3B and require $5B/year sales to be profitable…that's at least 5M wafer starts and almost 5B 1cm die /per year. A die (f=90nm) has 100-200 M tx /cm^2.
- Ultimately at O($2000) per wafer that's $2/ cm^2 or 100M tx. So how to use them?

Area and Cost

Is efficient use of die area important?

Is processor cost important?

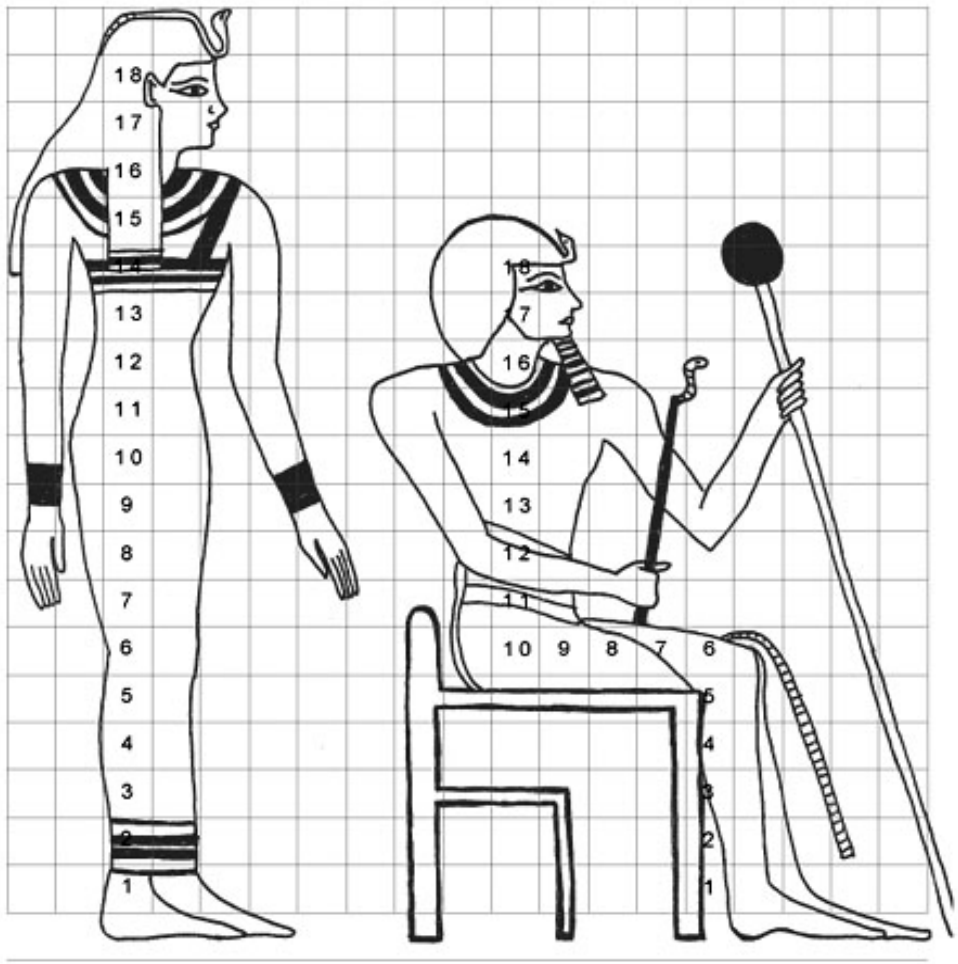
NO, to a point – server processors (cost dominated by memory, power/ cooling, etc.)

YES – everything in the middle.

NO – very small, embedded die which are package limited. (10-100Mtx/die)

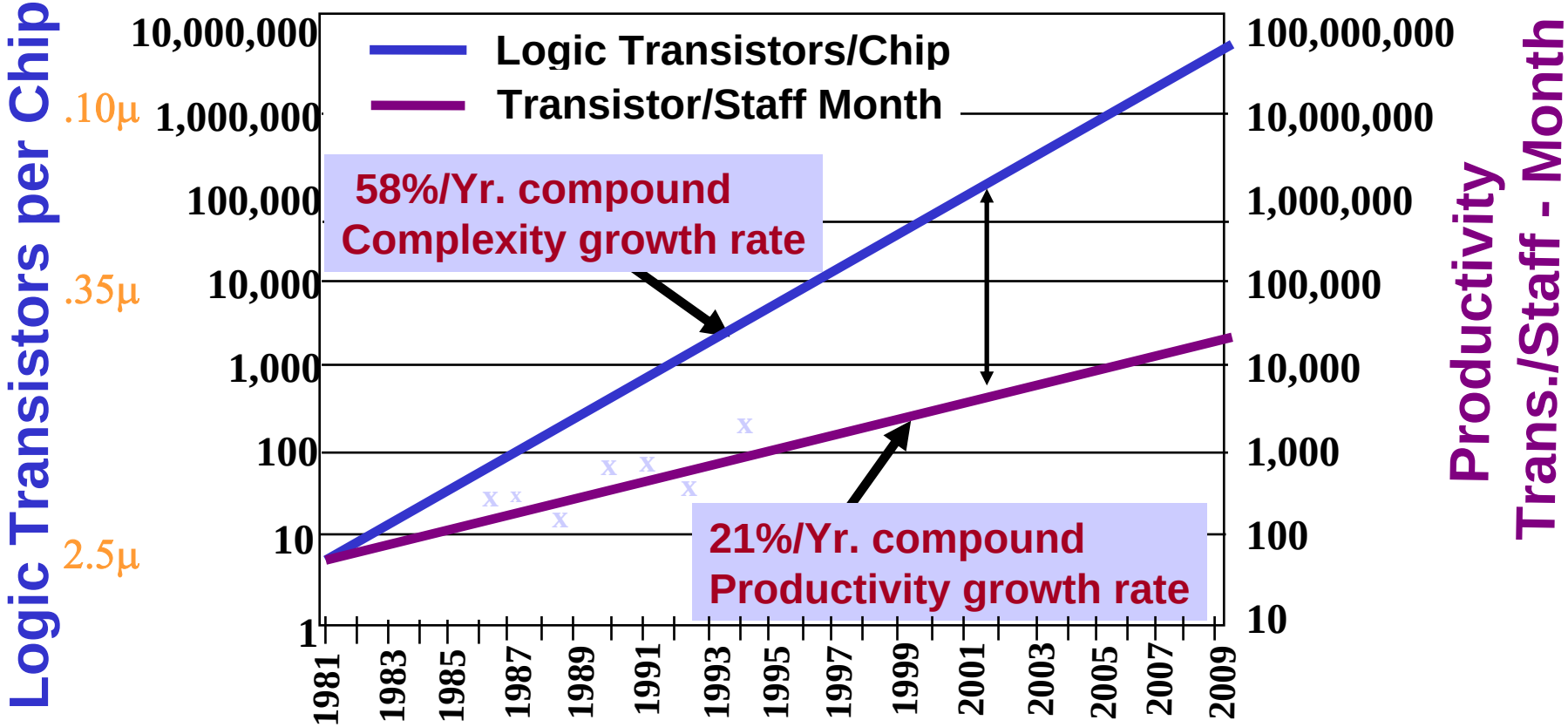
But it takes a lot of effort to design a chip

world's first CAD system?



© Canadian Museum of Civilization Corporation

Design time: CAD productivity limitations favor soft designs



Source: S. Malik, orig. SEMATECH

Time

www.snowder.com

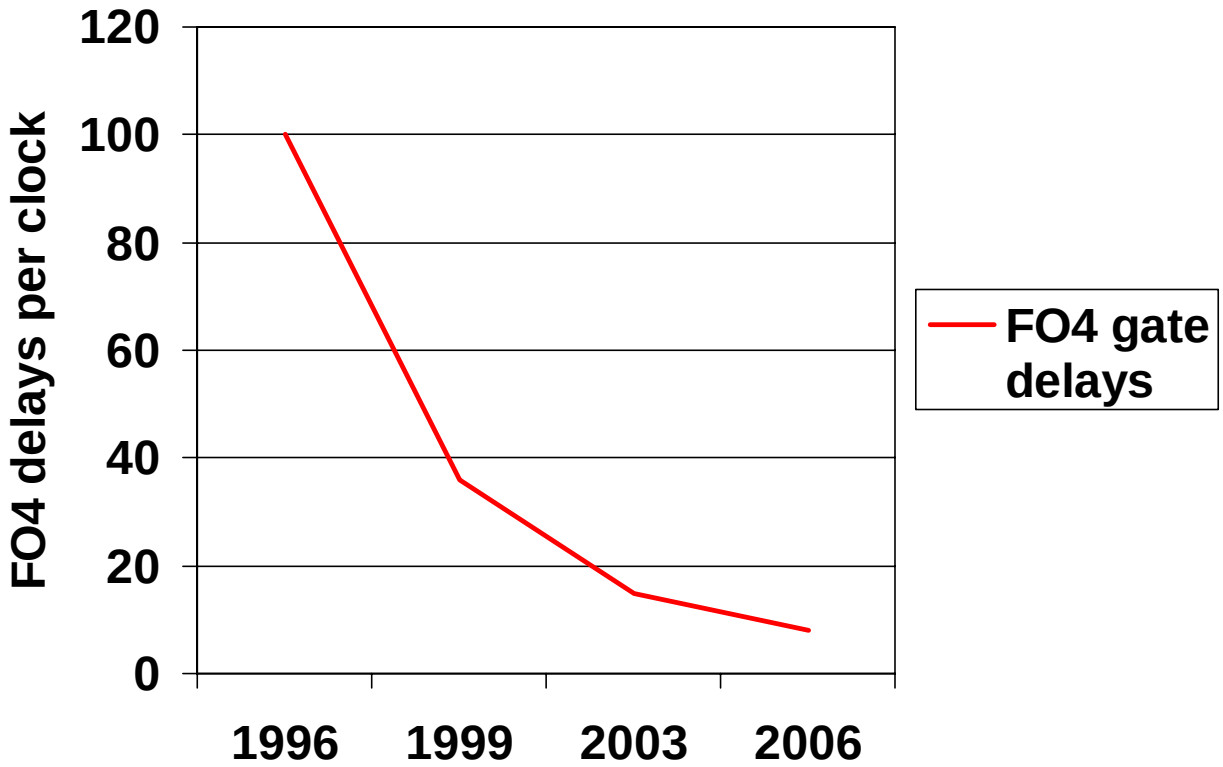
High Speed Clocking

Fast clocks are not primarily the result of technology scaling, but rather of architecture/logic techniques:

  - Smaller pipeline segments, less clock overhead

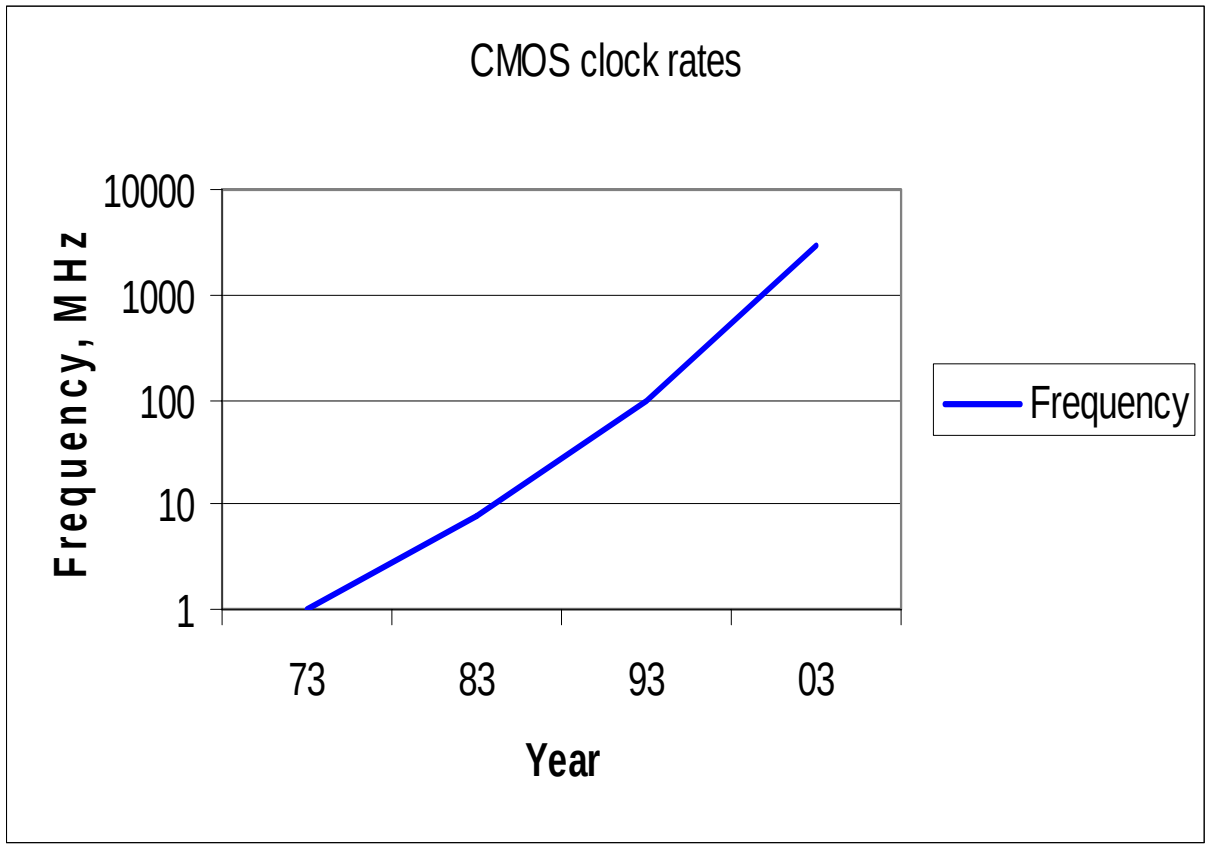
- Modern microprocessors are increasing clock speed more rapidly than anyone (SIA) predicted…fast clocks or *hyperclocking* (really short pipe segments)
- But fast clocks do not by themselves increase system performance

Change in pipe segment size



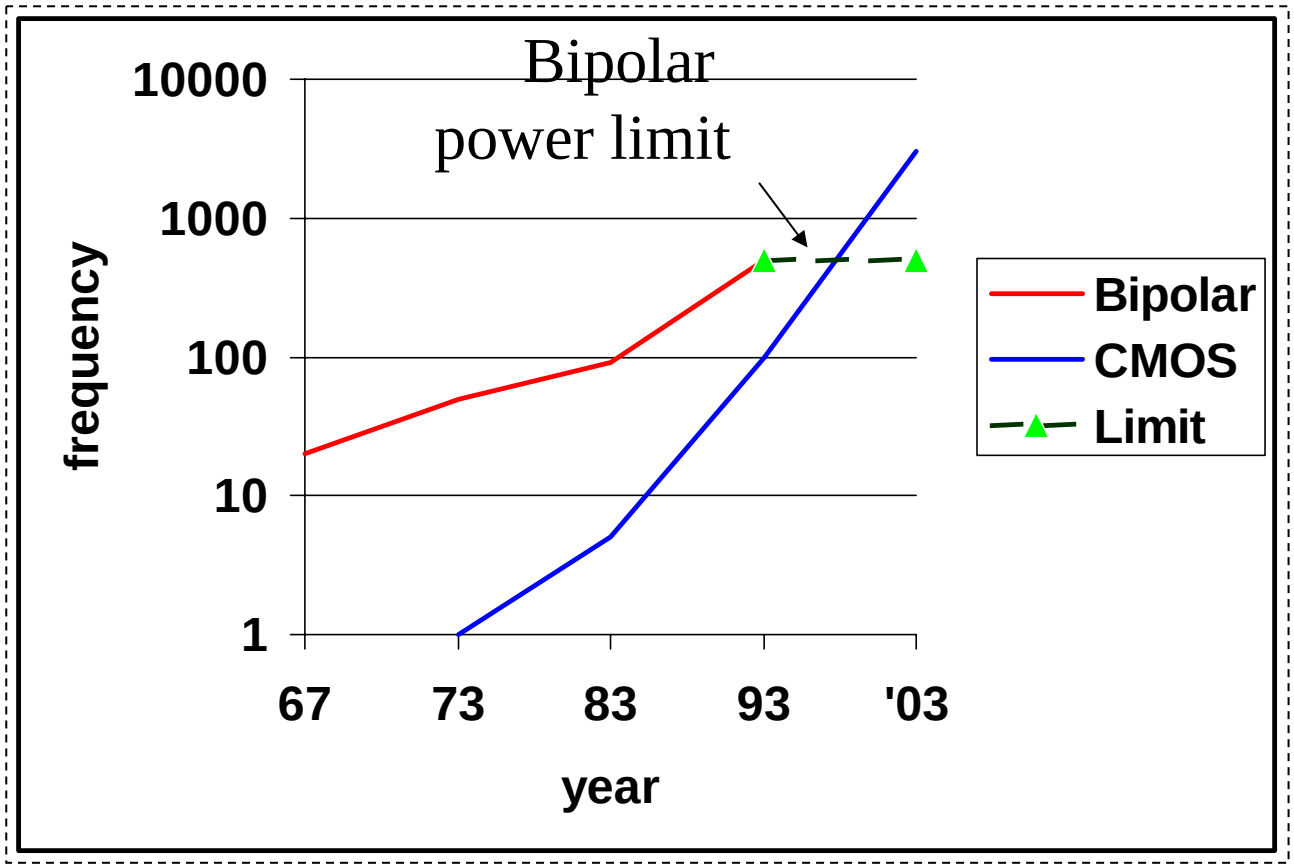
M.S. Hrishikesh et al., "The Optimal Logic Depth Per Pipeline Stage is 6 to 8 FO4 Inverter Delays." 29th ISCA: 2002.

CMOS processor clock frequency

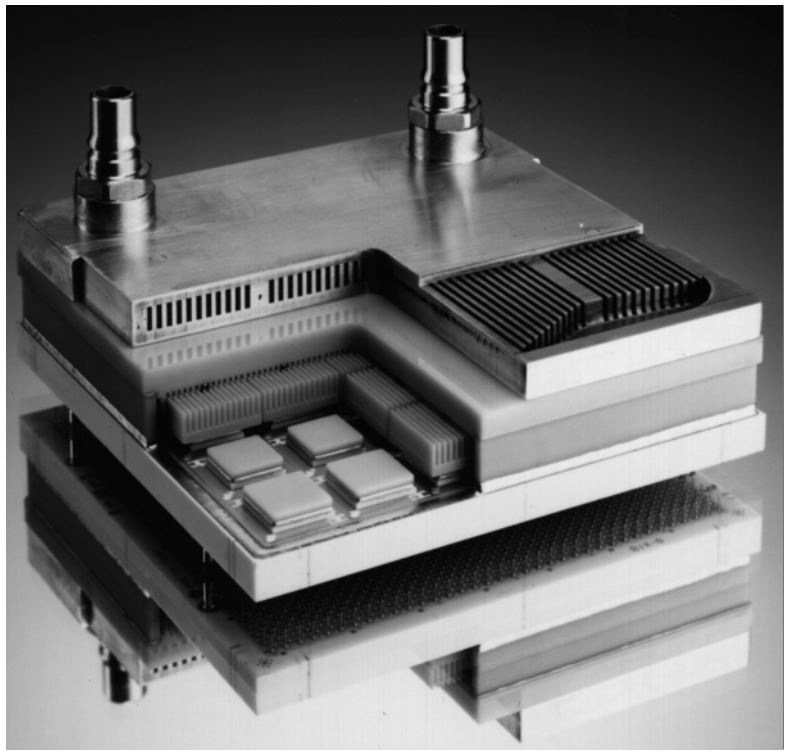


MHz History Chart: see Mac Info Home Page .

Bipolar and CMOS clock frequency



Bipolar cooling technology (ca ’91)



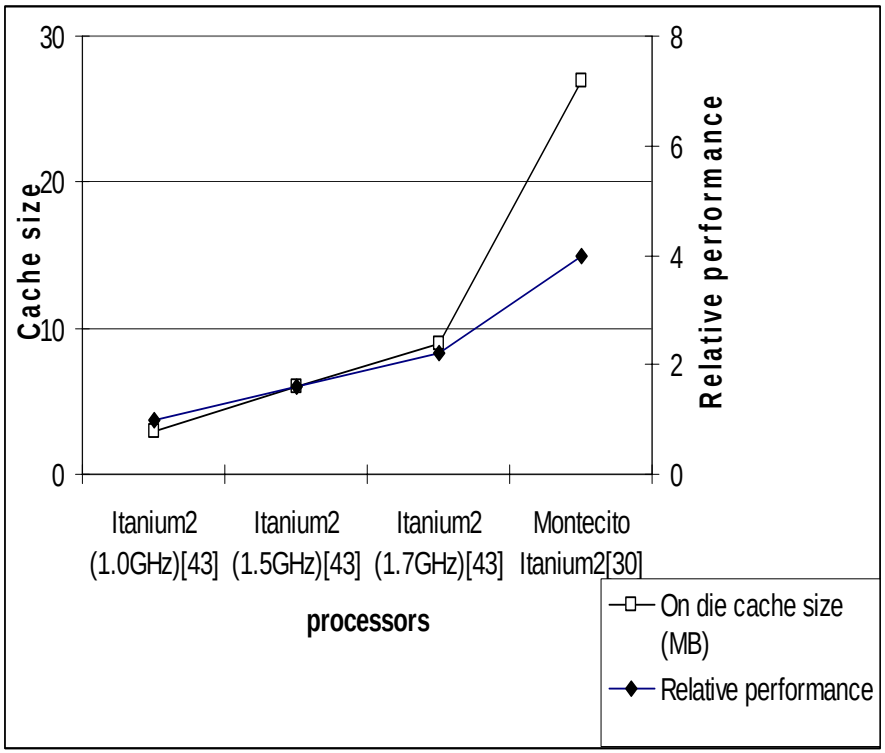
Hitachi M880: 500 MHz; one processor/module, 40 die sealed in helium then cooled by a water jacket. Power consumed: about 800 watts per module

F. Kobayashi, et al . “Hardware technology for Hitachi M-880.” Proceedings Electronic Components and Tech Conf., 1991.

Translating time into performance: scaling the walls

- ***The memory "wall":*** performance is limited by the predictability & supply of data from memory. This depends on the access time to memory and thus on wire delay which remains constant with scaling.
- But (so far) significant hardware support (area, fast dynamic logic) has enabled designers to manage cache misses, branches, etc reducing memory access.
- There's also a ***frequency*** (minimum segment size) and related ***power "wall"***. Here the questions is how to improve performance without changing the segment size or increasing the frequency.

Missing the memory wall



Cache size and performance for the Itanium processor family

- Itanium processors now have 9MB of cache/die, moving to > 27 MB
- A typical processor (in 2004) occupies < 20% of die, moving to 5%.
- Limited memory BW, access time (cache miss now over 300~)
- Result: large cache and efforts to improve memory & bus

S. Rusu, et al, "Itanium 2 Processor 6M: Higher Frequency and Larger L3 Cache." IEEE Micro, March/April 2004.

Power

- Old rough an’ ready, Zachary Taylor

Power: the real price of performance

$$P_{\text{total}} = \frac{C \cdot V^2 \cdot \text{freq}}{2} + I_{\text{leakage}} \cdot V + I_{sc} \cdot V$$

While V_{dd} and C (capacity) decrease, frequency increases at an accelerating rate thereby increasing power density

As V_{dd} decreases so does V_{th}; this increases $I_{leakage}$ and static power. Static power is now a big problem in high performance designs.

Net: while increasing frequency may or may not increase ***performance*** - it certainly does increase ***power***

Power

- Cooled high power: >70 W/ die
- High power: 10- 50 W/ die … plug in supply
- Low power: 0.1- 2 W / die.. rechargeable battery
- Very low power: 1- 100 mW /die .. AA size batteries
- Extremely low power: 1- 100 micro Watt/die and below (nano Watts) .. button batteries
- No power: extract from local EM field,

  …. O (1μW/die)

Achieving low power systems

By V_{dd} and device scaling $$\frac{freq_2}{freq_1} = \sqrt[3]{\frac{P_2}{P_1}}$$

- By scaling alone a 1000x slower implementation may need *only* 10^{-9} as much power.
- Gating power to functional units and other techniques should enable 100MHz processors to operate at $O(10^{-3})$ watts.
- Goal: $O(10^{-6})$ watts…. Implies about 10 MHz

Extremely Low Power Architecture (ELPA), ***O*** (μW), getting there….

1) Scaling alone lowers power by reducing parasitic wire and gate capacitance.
2) Lowering frequency lowers power by a cubic
3) Slower clocked processors are better matched to memory reducing caches, etc
4) Asynchronous clocking and logic eliminate clock power and state transitions.
5) Circuits: adiabatic switching, adaptive body biasing, etc
6) Integrated power management

ELP Technology & Architecture

- ***ELPA Challenge***: build a processor & memory that uses 10^{-6} the power of a high performance MPU and has 0.1 of the state of the art SPECmark (ok, how about .01?).

Study 1 StrongARM: 450 mW @ 160MHz

- Scale (@22nm) 450mW becomes 5 mW
- Cubic rule: 5mW becomes 5μw@ 16MHz
- Use 10x more area to recover performance
- 10x becomes 3-4x memory match timing.
- Asynchronous logic may give 5x in perf.
- Use sub threshold (?) circuits; higher V_T
- Power management

Study 2 the Ant

- 10 Hz, 0.1-1μW 250k neurons; 3D packaging, About ¼ mm^3 or about 1 $micron^3$ per neuron
- How many SPECmarks?

And while we're at it, why not an EHP challenge?

- Much more work done in this area but limited applicability
  - Whole earth simulator (60TFlops)
  - Grape series (40TFlops)
- General solutions come slowly (like the mills of the Gods)
  - Streaming compilers
  - New mathematical models

And don’t forget reliability!

Computational Integrity

Design for

- Reliability
- Testability
- Serviceability
- Process recoverability
- Fail-safe computation

Summary

- Embedded processors/SOC is the major growth area with obvious challenges:

1) 10x speed without increasing power

2) 10^{-6} less power with the same speed

3) 100x circuit complexity with same design effort.

- ***Beyond this the real challenge is in* system design: *new system concepts, interconnect technologies, IP management and design tools*.**